AUTONOMOUS NAVIGATION OF MOBILE ROBOT USING MODULAR ARCHITECTURE FOR UNSTRUCTURED ENVIRONMENT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Industrial Design

By

Saptarshi Mukherjee
(Roll: 213ID1366)

Department of Industrial Design
National Institute of Technology
Rourkela-769 008, Orissa, India
August 2009

AUTONOMOUS NAVIGATION OF MOBILE ROBOT USING MODULAR ARCHITECTURE FOR UNSTRUCTURED ENVIRONMENT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Industrial Design

By

Saptarshi Mukherjee

Under the supervision of

Prof. B.B. Biswal

Department of Mechanical Engineering
National Institute of Technology
Rourkela-769 008, Orissa, India
August 2009

National Institute of Technology
Rourkela-769 008, Orissa, India

CERTIFICATE

This is to certify that the work in the thesis entitled, "**Autonomous navigation of mobile robot using modular architecture for unstructured environment"** submitted by **Mr. Saptarshi Mukherjee** in partial fulfilment of the requirements for the award of **Master of Technology Degree** in the Department of Industrial Design, National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the work reported in this thesis is original and has not been submitted to any other Institution or University for the award of any degree or diploma.

He bears a good moral character to the best of my knowledge and belief.

Place: NIT Rourkela
Date:

Prof. (Dr.) Bibhuti Bhusan Biswal
Professor
Department of Industrial Design
National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

For each and every new activity in the world, the human being needs to learn or observe from somewhere else. The capacity of learning is the gift of GOD. To increase the capacity of learning and gaining the knowledge is the gift of GURU or Mentor. That is why we chanted in Sanskrit "*Guru Brahma Guru Bishnu Guru Devo Maheswara, Guru Sakshat Param Brahma Tashmey Shree Guruve Namoh*". That means the Guru or Mentor is the path of your destination.

The author first expresses his heartiest gratitude to his guide and supervisor *Prof. (Dr.) Bibhuti Bhusan Biswal*, Professor and Head of the Department of Industrial Design for his valuable and enthusiastic guidance, help and encouragement in the course of the present research work. The successful and timely completion of the research work is due to his constant inspiration and extraordinary vision. The author fails to express his appreciation to him.

The author is very much obliged to *Prof. (Dr.) Angel Valera Fernanadez,* Professor of the Department of *Automatica Y Informatica* (*Universitata Poletecnica de Valencia*) for his continuous encouragement and support.

The author is thankful to *Prof. (Dr.) Rajik Khan*, Assistant Professor of Industrial Design and *Prof. (Dr.) B.B.V.L Deepak*, Assistant Professor of Industrial Design, NIT Rourkela, for their support during the research period.

The help and cooperation received from the author's friend-circle, staff of Department of Training and Placement, staff of Department of Industrial Design is thankfully acknowledged.

Last but not the least, the author is forever indebted to his parents understanding and moral support during the tenure of his research work.

Saptarshi Mukherjee

ACKNOWLEDGEMENT

For each and every new activity in the world, the human being needs to learn or observe from somewhere else. The capacity of learning is the gift of GOD. To increase the capacity of learning and gaining the knowledge is the gift of GURU or Mentor. That is why we chanted in Sanskrit "*Guru Brahma Guru Bishnu Guru Devo Maheswara, Guru Sakshat Param Brahma Tashmey Shree Guruve Namoh*". That means the Guru or Mentor is the path of your destination.

The author first expresses his heartiest gratitude to his guide and supervisor *Prof. (Dr.) Bibhuti Bhusan Biswal*, Professor and Head of the Department of Industrial Design for his valuable and enthusiastic guidance, help and encouragement in the course of the present research work. The successful and timely completion of the research work is due to his constant inspiration and extraordinary vision. The author fails to express his appreciation to him.

The author is very much obliged to *Prof. (Dr.) Angel Valera Fernanadez*, Professor of the Department of *Automatica Y Informatica* (*Universitata Poletecnica de Valencia*) for his continuous encouragement and support.

The author is thankful to *Prof. (Dr.) Rajik Khan*, Assistant Professor of Industrial Design and *Prof. (Dr.) B.B.V.L Deepak*, Assistant Professor of Industrial Design, NIT Rourkela, for their support during the research period.

The help and cooperation received from the author's friend-circle, staff of Department of Training and Placement, staff of Department of Industrial Design is thankfully acknowledged.

Last but not the least, the author is forever indebted to his parents understanding and moral support during the tenure of his research work.

Saptarshi Mukherjee

Abstract

This article proposes a solution for autonomous navigation of mobile robot based on distributed control architecture. In this architecture, each stage of the algorithm is divided into separate software modules capable of interfacing to each other to obtain an effective global solution. The work starts with selection of suitable sensors depending on their requirement for the purpose and for the present work a stereo vision module and a laser range finder are used. These sensors are integrated with the robot controller via Ethernet/USB and the sensory feedbacks are used to control and navigate the robot. Using the architecture, an algorithm has been developed and implemented to intelligently avoid dynamic obstacles and optimally re-planning the path to reach the target location. The algorithm has been successfully tested with a Summit_XL mobile robot. The thesis describing the present research work is divided into eight chapters. The subject of the topic its contextual relevance and the related matters including the objectives of the work are presented in Chapter 1. The reviews on several diverse streams of literature on different issues of the topic such as autonomous navigation using various combinations of sensors networks, SLAM, obstacle detection and avoidance etc. are presented in Chapter 2. In Chapter 3, selected methodologies are explained. Chapter 4 presents the detail description of the sensors, automobile platform and software tools used to implement the developed methodology. In Chapter 5, detail view of the experimental setup is provided. Procedures and parametric evaluations are given in chapter 6. Successful indoor tests results are described in chapter 7. Finally, Chapter 8 presents the conclusion and future scope of the research work.

CONTENTS

TITLE	PAGE

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1. Introduction

1.1. Background

Advances in computer technology, wireless communications and sensing technologies are creating considerable interest in the field of robotics and mechatronics. A few years ago the production and programming of robots was focused on providing a robust and closed design where the robot successfully fulfil the mission to perform a specific task. In recent years, there is a high interest by the research community to extend the use of robotics in various fields such as medicine (Shibata, 2012), (Ringer et al., 2010), assistance to people (Körtner et al., 2012), (Mukai et al., 2010) rescue operations (Nagatani et al., 2011) missions in dangerous or inaccessible environments (Toit & Burdick, 2012), etc. These tasks are typically performed in uncontrolled environments and are hardly predictable. However these fields of applications have little or no scope to random or predefined controlled motion of the device. These tasks/activities must have precisely operated robotic motions in conjunction with appropriate sensory feedback. At present, most applications related to robotics require a complex control because they involve a large number of elements. In the control scheme the typical elements are sensors, actuators or the controller itself, among other elements. Hence there is a growing trend in component-based software development. At the same time, there has also been a natural evolution and maturation of the current software development, in which object-oriented programming is fully extended, to the point where object and component have a similar meaning, emphasizing interoperability and reusability (Shaw & Clements, 2006).

Component-based software development (CBSD) can be defined as an alternative design that promotes the reusability of software components, making it easier to develop other systems from existing ones. A software component is an abstraction unit with well-defined interfaces in order to improve the development time and maintenance (Alonso, Pastor, Sánchez, Álvarez, & Vicente-chicote, 2012) (Tang, Mu, Kwong, & Luo, 2011). On the other hand, another very important aspect of software development is the cost (not only in terms of resources, but time too). This line of research with respect to CBSD has not evolved much in software engineering yet, and it is still a very immatured area (Wijayasiriwardhane, Lai, & Kang, 2011). Therefore it is very complicated to estimate the cost depending on the component's life cycle or the number of components. The use of control-based software development is advantageous over conventional software development, because of its in-built features leading to fast tracking of the execution flow program, easy creation of new components with new functionalities, scalability in a developed scheme, reusability of code, and also making it possible to build an unlimited number of components. Fortunately, in the field of robotics there is a range of different middleware or frameworks that facilitate the use of CBSD. Middleware provides a common programming abstraction across a distributed system (Gill, Smart, & Louis, 2002) in order to achieve a number of objectives such as portability, reliability and managing complexity (Gill et al., 2002).

The robots control middlewares (such as Orocos (Bruyninckx, 2001), Orca (Makarenko, Brooks, & Kaupp, 2006), Player (Gerkey, Vaughan, & Howard, 2003), Brics (Bischoff et al., 2010) or ROS (Quigley et al., 2009)), provide a set of generic tools and libraries, drivers and methodologies or algorithms, to facilitate the integration and development of robot software. In traditional sequential programming various stages are solved step by step in order to make up a target. In recent years, such architectures have evolved towards an increasingly modular structure (Essen, Hirvonen, & Kuikka, 2014). Modular systems are considered as complex distributed systems, composed of several independent modules to perform tasks such as the integration between hardware and software (B. Basso, 2010). The system's sensors, actuators and controllers become individualized instruments which can be controlled by these modules in cooperative manner to achieve specific tasks. The structure of the system allows easier scalability, reusability, flexibility and also has the ability to interact with other systems via communication networks in a simple and transparent manner. In the field of mobile robotics area, the problem of autonomous vehicle navigation is one of the most studied subject (Nakhaeinia and Tang. 2011), (Sales et. al. 2011). Major application areas of autonomous navigation for mobile robots are development of sentinel for enclosures surveillance, monitoring and finding objects of interest in dangerous scenarios or dangerous facilities, exploring unknown or hostile environments etc. The autonomous navigation includes different sub problems such as the estimation of correct location of the vehicle within a scenario, collection and processing of information about the environment, planning the path from the initial to the target point, monitoring strategic path, avoidance of potential obstacles that may stand in its path, etc. (Moghadam & Wijesoma, 2008) proposed an improved path planning technique by the help of stereo vision and laser data. For irregular shaped obstacles 3D point cloud data and disparity image are used and finally a 2D map is generated whereas for regular objects 2D cost map is generated using a laser scanner. Later those two maps are fused for successful path planning. An optical flow based autonomous navigation and obstacle avoidance is proposed by (Dur, 2009) where a laser range finder is used to sense the obstacle distance from the robot by the help of triangulation law. Consecutive image frames from the stereo camera are processed to determine the relative change of location of the detected object (optical flow) and finally the generated data are supplied to Artificial Neural Network (ANN; Levenberg- Marquardt learning algorithm) to train the model. (Moon, Kim, & Kim, 2007) proposed a navigation system model for AGV using laser and camera. The laser model is used to detect the obstacle. Laser data reflected from same object are clustered to plot on the map. The images from the vision are used to detect the lane using lane detection algorithm.

1.2. Objectives

The present research proposes a solution to the problem of autonomous navigation of mobile robots from a modular point of view based on the use of a distributed architecture and advanced sensors. The approach offers an optimized use of individual sensors and their processing costs. Major objectives of this work are use of stereo vision to detect the obstacles and simultaneously extract information about them. Later that information will be used to generate the best possible path. While travelling along the generated path the laser scanner will take care of the dynamic obstacle. The process will repeat until the mobile robot reaches the target point. There are many approaches and proposals to solve navigation issues of autonomous mobile robot. However, carrying out its implementation using real robots with traditional approaches requires constant translations of code depending on the language supported by each platform as well as code modifications if different hardware and software options are used.

CHAPTER 2

2. Literature Survey

2.1. Overview

In the field of autonomous navigation various works had already been done. Major landmark works are tabularised in table 2.1. Later total work is categorized according the various combination of sensors used. Major sensors considered are distance measurement unit, Omni-directional camera and stereo camera. Major sensor networks considered are various distance measurement units, various omni-directional cameras along with the stereo setup, and distance measurement units along with the camera setup.

2.2. Major works done so far on autonomous navigation

Table 2.1: Key works done in the field of autonomous navigation

Sl. No.	Title	Author	Source	Sensors & Robots	Remark
1.	Neural Fields for Complex Behavior Generation on Autonomous Robots	Mohamed Oubbati	Proc. of International Joint conference on Neural Networks (2009)	STANISLAV, an RWI B21 robot, sonar sensors (24), laser Sensor(1)	A neural field model designed to perform three important and complex tasks for mobile robots such as target acquisition, obstacle avoidance, and sub target selection. The advantage of neural field over potential field was also explained. Successful indoor tests were performed.
2.	Improvement on Obstacle Avoiding Ability Based on Laser Range Finder	Fujun He	Proc. of IEEE International Conference on Mechatronics and Automation (2010)	Auto mobile platform, Laser Measurement System-200	A three-layered sub-goal based obstacle avoiding policy was proposed for fast and arbitrarily moving obstacles. Before having any physical contact with the obstacle it can estimate the collision time and position using tracking algorithm. According to the collision estimation or obstacle category it will choose the best policy among the proposed three. Since the Laser Range finder wouldn't be able to sense the longitudinal size of the object a larger safety distance region is set.
3.	Lidar Based Off-	Jacoby Larson	International IEEE	iRobot Packbot,	A terrain analysis method for outdoor environment using

	road Negative Obstacle Detection and Analysis		Conference on Intelligent Transportation Systems (2011)	UTM-30LX lidar sensor	laser range finder was proposed. The 3D point cloud data generated from 3D Laser Range Finder is used to detect the negative obstacle with the help of two special methods, Negative Obstacle Detector (geometry based method) and Support Vector Machine.
4.	Marathoner Tracking Algorithms for a High Speed Mobile Robot	Eui-Jung Jung	Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (2011)	Marathon Following Robot (MRF), UTM-30LX Laser Range Finder	A tracking algorithm was proposed for mobile robot that will follow the human target at higher speed and in unstructured outdoor environment. Developed algorithm is implemented using a differential drive robot and a laser sensor. Human wrist part was used as the best tracking point and one obstacle avoidance algorithm was proposed using the relative velocity of the robot and the obstacle.
5.	Probabilistic Obstacle Detection Using $2\frac{1}{2}$ Terrian Map	Gregory Broten	Proc. of Conference on Computer and Robot Vision (2012)	Raptor Unmanned Ground Vehicle, SICK Laser Scanner (2),	In this paper, probabilistic obstacle detection method was proposed by using the Laser Range Data from Laser Scanner and $2\frac{1}{2}$. Major advantage of the proposed method is it eliminates the need of multiple data source. Thereby reducing the sensor cost as well as data processing cost. Simulation using Gazebo, and real world outdoor test.
6.	Multi Lidar System for fast obstacle detection	Inwook Shim	Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence (2012)	Robotic platform, SICK LMS 151 (2),UTX-30LX (2), IBEO LUX (1),	A high-speed obstacle detection method was proposed using the gradients of points in outdoor environment. In order to increase the accuracy of the result the laser sensors were reconfigured using planer rule for mostly outdoor robotic platform. Processing time as well as the accuracy of the system was evaluated during outdoor test.
7.	Development of a Laser-Range-Finder-	Eui-Jung Jung	IEEE/ASME transactions on Mechatronics	Marathoner Service Robot” (MSR), UTM-30LX	Human detection as well as obstacle avoidance algorithm was proposed. Human detection was done by laser range finder. In order to make

	Based Human Tracking and Control Algorithm for a Marathoner Service Robot			(human), URG-04LX (obstacle)	the system more robust multiple (539) human scan data are feed into the system to train the SVDD model. Moving obstacles are avoided by providing a weight radius to each obstacle according to the relative velocity of the obstacle with the robot. Moreover dynamic obstacle avoidance algorithm was used to create a smooth position vector between the robot and the shortest path around the obstacle.
8.	Towards Bringing Robots into Homes	MarkusVincze	Jr. of ISR / ROBOTIK (2010)	Stereo Vision	An object recognition and avoidance method was proposed using stereo camera. Localization problem was solved by matching the free space to the loaded map. The advantage of the proposed system above the laser module is it can detect object of any height.
9.	Learning Long-Range Vision for Autonomous Off-Road Driving	Raia Hadsell	Jr. of Field Robotics (2009)	LAGR (Learning Applied to Ground Robots) mobile robot, Stereo Camera	A self-supervised stereo learning process was discussed. The proposed method can accurately classify complex terrain at distances up to the horizon. The reason behind this accurate result is due to the self-supervised training data that are generated on every frame.
10.	Binocular Stereo Vision Based Obstacle Avoidance Algorithm for Autonomous Mobile Robots	Saurav Kumar	Proc. of International Advance Computing Conference (2009)	"Lakshya"- an IGV robot platform, BumbleBee Stereo Vision camera	A Binocular Stereo vision system based autonomous navigation method was proposed. In order to reduce the computational cost instead of process the whole disparity image the 3D point cloud data is being used. Total workspace in front of the robot was divide into grids and allocated with 3 logics (obstacle or free or unknown).
11.	A Probabilistic Framework for Stereo-Vision Based 3D	Jeremy Ma	Proc. of International Conference on Robotics and Automation (IEEE 2010)	Evolution Robotics ER-1 robot, BumbleBee2 stereo color camera	An efficient way for 3D object search, 6D pose estimation of the detected object and finally path planning was done. The total search method was divided into two basic steps local search (3D SIFT

	Object Search with 6D Pose Estimation				features) and global search. Total method was supervised by Bayesian recursion methods, and during the process it was being updated.
12.	Vision Based Obstacle Avoidance And Target Tracking for Autonomous Mobile Robot	Masahiro Yamada	IEEE International Workshop on Advanced Motion Control (2010)	Automobile platform, STH-MDCS2 stereo camera pair, QCAM Pro 5000 web camera.	A novel target tracking and obstacle avoidance algorithm was proposed by using data from a pair of stereo camera and one Omni-directional monocular camera. Stereo camera pair was used to build an obstacle map with the help of V-disparity and monocular camera was used to keep track of the target by self-windowing technique. Moreover a switching strategy was also applied in between obstacle avoidance state and target tracking state.
13.	A local obstacle avoidance method for mobile robots in partially known environment	Chaoxia Shi	Jr. of Robotics and Autonomous Systems (2010)	Pioneer III robot (main), HIT II robot (dynamic obstacle), laser range finder and camera	A new method PBCM (prediction based curvature method) was proposed which inherits the quickness of BCM (beam curvature method), safety of LCM (lane curvature method). Moreover this method can avoid dynamic obstacle by its prediction model. Still some modifications are required for parameter selection and cases where there is a sudden appearance of dynamic obstacle.
14.	Development of a low-cost active 3D triangulation laser scanner for indoor navigation of miniature mobile robots	Guoqiang Fu	Robotics and Autonomous Systems (2012)	mobile robotic platform, miniaturized camera OV7670, laser diode LT-1-650-CL02	A low-cost active 3D triangulation laser scanner for indoor navigation of miniature mobile robots was presented and was implemented by moving both a camera and a laser diode together on the robot's movable part. 3D laser scanner projects its light on the obstacle. The camera captures the laser projected image and using triangulation law calculates the obstacle width and the gate way gap.
15.	Environment-Detection-and-	Jaewoong Choi	Jr. of Intelligent Transportation Systems	SNUCLE (automobile test platform),	A rural as well as off-road environment-detection-and-mapping algorithm was provided for autonomous

	Mapping Algorithm for Autonomous Driving in Rural or Off-Road Environment			CCD cameras, SICK Laser sensor, IBEO scanner.	vehicle. It consists of two parts and each part was being performed by the use of laser and camera respectively. Lane, pedestrian-crossing, and speed-bump detection are done by camera module. Obstacle detection and their localization were done by LIDAR.
16.	Visual servoing based mobile robot navigation able to deal with complete target loss	Wenhao Fu	Jr. of IEEE (2013)	Lina mobile robot, odometry sensor, Laser Range Finder, ACTi IP Camera.	A vision based autonomous robot navigation method along with obstacle avoidance was proposed. The obstacles were detected by the help of laser range finder. While avoiding the obstacle camera may lose the target point. In this case with the use of odometry data the robot can again realign towards the target point.
17.	Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance	Andrea Cherubini	IEEE Transactions on Intelligent Transportation Systems (2014)	CyCab vehicle (robot), B&W Marlin (F-131B) camera, SICK LD-MRS laser scanner.	A vision and laser based autonomous navigation method was proposed. Several key images were supplied beforehand. Later on those key images were used by the vision module to make the robot run in predefined path. Presences of sudden obstacles in the path were taken care by the laser scanner. In case of moving obstacle the relative velocity is estimated by means of Kalman Filter and used in the obstacle avoidance module.
18.	Sensor Fusion of Laser & Stereo Vision Camera for Depth Estimation and Obstacle Avoidance	Saurav Kumar	International Journal of Computer Applications (2010)	Lakshya‟- an IGV robot platform, URG-04LX LRF, BumbleBee Stereo Vision camera	A data fusion technique of laser range finder and stereo vision camera was proposed for depth estimation and navigation. 3D point cloud data from the stereo vision was scaled down to 2D vision map. The generated map was fused with the 2D cost map from the laser range finder. Finally the map needs to pass through the VFH+ for obstacle avoidance and navigation. Successful test were performed. Main focus of this work was to reduce the computational cost of fusion process.

2.3. Obstacle avoidance using distance sensors

(Chi & Lee, 2011) proposed a speed controller design using the combination of fuzzy logic and neural network. This system includes three neural controllers for local navigation, two neural networks for environmental recognition, and a fuzzy system for direction and speed control. In this experiment 24 ultrasonic range finders are used to collect the distance data and finally feed them to the fuzzy logic system. The system contains 77 rules of which 28 for Dead End Check, 34 for Direction Control, and 15 for Speed Control. In future the technique can be used for SLAM. Successful indoor tests were performed using the differential drive mobile robot.

(Jeong, Kim, & Kwak, 2009) provided an example of system construction and experimental results for an autonomous mobile robot applicable to indoor messy environments. Using PSD (Position Sensitive Detector) and LMS (Laser Measurement System) it calculates the distance between the robot and surrounding obstacles and finally leads to a suitable driving mode until it reaches the target point. Successful indoor tests were performed.

(Guo, Su, Wang, & Wu, 2009) proposed a new mobile robot design using laser range finder, ultrasonic sensor and IR sensor. Ultrasonic sensor was used for some typical obstacles like glass materials those weren't detected by laser range finder. Finally the sensory data were used to program the NI motion controller card so that it could follow the predefined trajectory along with avoiding the obstacle. Successful indoor tests were performed.

(Zhao, Wu, Lou, & Ogai, 2011) developed a novel mobile robot control method using only distance data from laser range finder. The distance as well as the location of the obstacle was determined using distance information of each laser angle. The sub goal location point was calculated using the previously calculated obstacle location. The process repeated until the robot reaches its final goal or target location. In future this information can be used for trajectory generation.

(Yu et al., 2010) proposed path planning and obstacle avoidance method using distance information provided by the Laser Radar. The method used improved the Artificial Potential Field (APF) method and all measurements were transferred into polar coordinate. Laser Radar location was assumed to be the center of the polar coordinate system and the robot-to-obstacle relative distance was measured. These measured results were used to adjust the robot velocity and direction in order to avoid obstacles. In future the developed method needs to implement in more complex scenario.

(C. L. Chen, Chou, & Lian, 2011) proposed and compared two target following methods. First is an intuitive method; it sets a pseudo goal further away than the actual target location and used obstacle avoidance algorithm to reach the pseudo goal. The second method used heuristic search to find a trajectory which can maximize target visibility and minimize the distance between robot and target at the same time. In future more complex scenarios need to be tested with the help of more advanced tracking algorithm.

(Kim & Kim, 2011) developed an overlapped ultrasonic sensor ring for obstacle detection in autonomous navigation. Set of low directivity ultrasonic sensors were placed along a circle of nonzero radius with equal spacing. Structural constrains were used to avoid non-empty overlap beam as well as excess beam. Positional uncertainties were also taken care by modifying the design parameter.

(Karambakhsh, Yousefi Azar Khanian, Meybodi, & Fakharian, 2011) proposed a Fuzzy based robot localization and navigation method. Laser data and sonar data were fused using Fuzzy based Kalman Filter to localize the robot in the map and thereby navigating the robot in an obstacle free manner. Successful simulator tests were performed.

(Arrichiello, Chiaverini, & Mehta, 2012) proposed an obstacle and collision avoidance algorithm for multi-robot distributed system. A localization algorithm was implemented for the navigation in indoor environment. Null-Space based Behavioral control (Behavior based technique) was used to make the robots navigate avoiding collision among them and using the laser obstacles was detected. Communication among the robots was done by AD-HOC network. Successful indoor tests were performed.

(Kang et al., 2012) proposed a new laser data clustering method. In the traditional data cluster methods the threshold values were fixed and were not much helpful for unstructured environment. In the proposed adaptive data clustering method threshold values will be obtained adaptively and effectively by probability event principle. Successful outdoor tests were performed.

(Zug, Penzlin, Dietrich, Nguyen, & Albert, 2012) performed comparison between the existing laser scanners and the new Kinect sensor for map building, localization and obstacle avoidance, The result concluded at the end is laser range finder provides way better result as compare to single laser.

(Wu et al., 2012) proposed a novel autonomous navigation method for snake like robot in obstacle free manner. The laser and sonar data were fused based on TS Fuzzy Neural Network and later used to detect the surrounding environment as well as taking proper control action to avoid the obstacle. Successful indoor test run were performed.

(Hoang et al., 2013) proposed a novel navigation and obstacle avoidance method by means of obstacle detection, path planning and trajectory tracking. Laser range finder was used to generate the 3D point cloud of the current environment. This 3D point cloud was modified to 2D data in order to generate the map and plan the suitable trajectory. Obstacle avoidance was done by the help of sonar data and modified Vector Field Histogram method. Successful indoor tests were performed.

(Voda, 2013) presented an algorithm for trajectory-tracking with obstacle avoidance for autonomous vehicles. The algorithm used Quintic equations to generate the obstacle free paths for the automobile platform. The generated trajectories were the composition of global trajectory provided and the local modified trajectory due to the presence of the obstacle. Successful indoor tests were performed.

(Obstacle et al., 2013) proposed the design of an emergency obstacle avoidance module with the help of laser range finder for mobile robot. Due to the sudden arrival of obstacle fast system response was also considered. Successful indoor tests were performed.

(Wei, Zeng, & Wu, 2013) proposed a novel obstacle detection technique using only large range sonar. At first vehicle odometery was incorporated in the sonar to convert the 1D distance information to 2D signal. Hungarian algorithm was used to identify the reflected signal from mutual object. Later these points were used to extract the obstacle feature.

(Danilo, 2014) proposed a laser based object detection and avoidance algorithm. By means of Density-based spatial clustering of applications with noise (DBSCAN) the obstacles are detected and then clustering analysis was done to generate the optimal path. Successful Simulation tests were performed.

(Cheam, Bakar, & Saman, 2014) proposed an object tracking algorithm with the help of low cost laser scanner and ultrasonic sensor. The proposed algorithm was successfully tested in follow-the-leader mode in indoor environment.

2.4. Obstacle avoidance using camera

(Huang, Fajen, Fink, & Warren, 2006) proposed a vision guided local navigation system using the steering potential function. The goal potential function is a parabolic bowl centered on the goal heading, and the potential function for each obstacle is a peak centered on the obstacle heading. These two potential functions were used to calculate the required angular velocity. Successful indoor tests were performed.

(Ess, Leibe, Schindler, & van Gool, 2009) introduced a new approach which jointly estimates camera position, stereo depth, object detections, and trajectories based only on visual information. At first the terrain and obstacles were identified. Based on this result object trajectory was estimated and with the help of tracking algorithm the future motion was also predicted for dynamic obstacle. Using above mentioned results the optimal path could be planned in any dynamic complex situation.

(Cherian, Morellas, & Papanikolopoulos, 2009) proposed a novel 3D reconstruction technique using a single camera and Markov Random Field (MRF) method. The 3D construction was done defining a coordinate frame with respect to environment. The coordinate values were used to localize the landmark in the map. A texture based segmentation algorithm was also applied to isolate the terrain from the objects. Successful tests were performed using outdoor images.

(Hwang S. N., 2010) proposed a collision avoidance method using vision module where the Laser Range Finder or Ultrasonic sensor will not work. In case of a slope road the stereo vision module will detect the corner points as well as the slope angle using the Harris Corner Detection and Randomized Hough Transform respectively. Later those corner points will be used to locate and avoid the obstacles. Successful indoor and outdoor tests were performed.

(Son, Kim, & Choi, 2010) proposed a novel method for 3D object detection and modeling using the range data from a 3D range sensing camera. The total method consisted of data acquisition, pre-processing, object segmentation, and finally generation of 3D model. The speed and accuracy of the generated model was tested in real time.

(Aggarwal, Kukreja, & Chopra, 2010) developed an autonomous navigation algorithm using images only from monocular vision camera. A depth map was created by allocation of every pixel of the image as either ground or obstacle. After this a virtual triangular obstacle was being created in the depth map, and the robot was directed to follow the edge in order to avoid the obstacle. With this algorithm the robot will take a path where the obstacle density is comparatively less. Successful indoor tests were performed.

(Perrollaz, Yoder, & Laugier, 2010) proposed a novel method for occupancy grid generation with the help of stereo vision. The basic advantage is it reduces the cost for processing a huge point cloud data. Using the U-disparity image the obstacle and road detection was done. Later this disparity-space occupancy grid was transformed into a Cartesian space occupancy grid including the reduction of discretization effect. Successful test were performed using real time images.

2.5. Obstacle avoidance using laser and camera

(C. C. Chen, Lu, Chang, Tsai, & Tang, 2009) proposed a new method for the calculation of obstacle distance from the robot which used only a single camera and a laser projector. The projected laser point gap was calculated using image processing tools and pixel gap value was used in the triangulation law to calculate the distance of the obstacle. Successful Indoor tests were performed.

(Csaba, Somlyai, & Vamossy, 2011) proposed an autonomous navigation system using laser diode and webcam. The projected laser light along with the background was captured by webcam. Later using image processing tools the objects were detected. Intensity threshold technique and triangular method were used to determine various properties of the detected obstacle. In this method objects at a higher altitude or with glass surface will create a problem during detection. Indoor test result is way better then outdoor test result due to the light conditioning.

(Lin, Lin, Liu, & Chen, 2013) proposed a novel autonomous navigation system based on monocular vision and laser based. Object recognition was done by the vision model. Kalman filter technique was used to set the threshold value according to the current color condition. Moreover instead of searching the whole image pre predicted regions were searched in the new window to reduce the computation time. Successful indoor tests were performed.

(Einhorn, Schröter, & Gross, 2009) developed a novel technique for outlier free scene reconstruction with the help of sequence of images. The reconstruction of the scene was done with the help of Extended Kalman Filter and can detect obstacle those were not visible by the laser scanner. Through successful indoor tests it was established that use of laser range finder along with the camera can increase the detection rate.

(Moghadam & Wijesoma, 2008) proposed an improved path planning technique by the help of stereo vision and laser data. For irregular shaped obstacles 3D point cloud data and disparity image was used and finally a 2D map was generated. For regular objects using a laser scanner 2D cost map was generated. Later those two maps were fused for successful path planning. Successful indoor tests were performed.

(Dur, 2009) developed an optical flow based autonomous navigation and obstacle avoidance method. Laser range finder was used to sense the obstacle distance from the robot by the help of triangulation law. Consecutive image frames from the stereo camera was processed to determine the relative change of location of the detected object (optical flow). The generated data was supplied to Artificial Neural Network (ANN) (Levenberg- Marquardt learning algorithm) to train the model. Successful simulation test runs and optical flow calculation were performed with real time image pairs.

2.6. Summary

All the sensor network combinations are considered and it’s summarized that the best possible combination is stereo camera and distance measurement unit together. Obstacles missed by the laser unit will be detected by the camera unit. Since camera takes longer time to process the generated image, so the dynamic obstacles will be detected by the laser module which requires lesser processing time. Basically the drawbacks of one unit will be covered by the other and vice versa.

CHAPTER 3

3. Methodology

3.1. Overview

Major key issues in the field of autonomous navigation are generation of obstacle free path in-order to progress and detection of various dynamic obstacles that could occur in its path. Generation of obstacle free path was done by the stereo camera setup and dynamic obstacles were avoided by the help of laser scanner. Total methodology of the above mentioned task is explained below with the help of a flow chart as shown in figure 3.1.

3.2. Flow chart

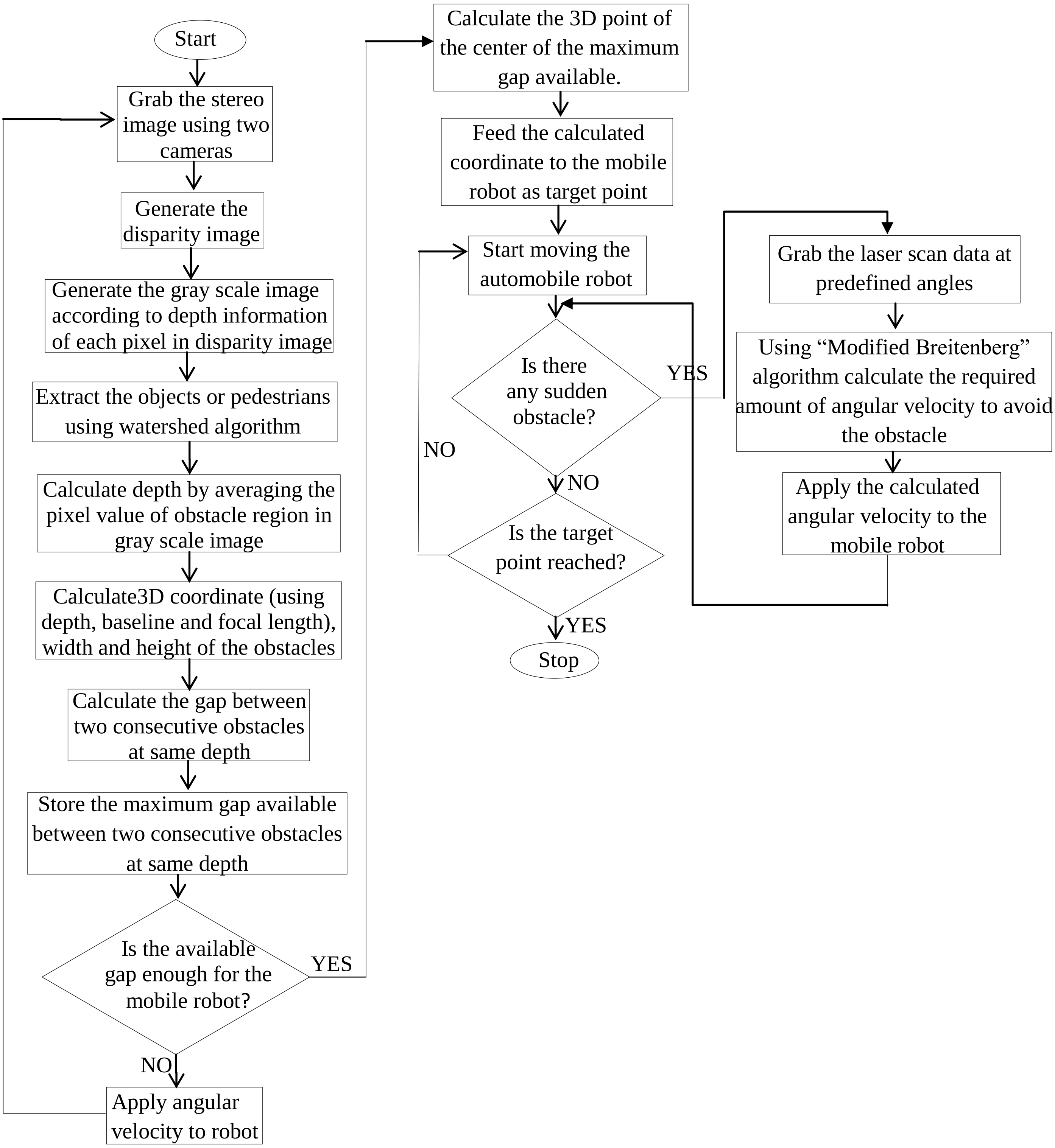


Fig: 3.1 Flow chart generalized obstacle avoidance for mobile robot

3.3. Task performed by stereo vision

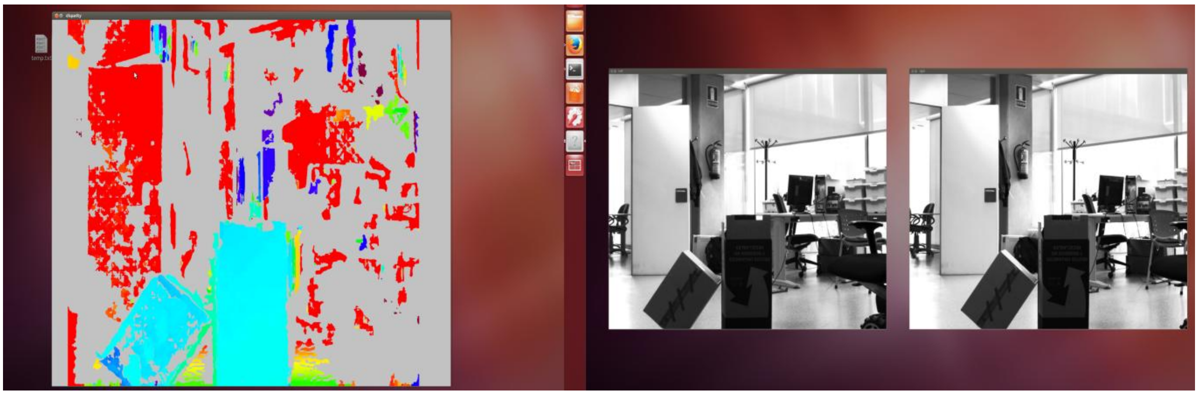
Generation of disparity image

Before generating the disparity image the stereo setup needs to be calibrated with the help of a checked board with known dimension. For calibration a ROS package *"camera_calibration"* has been used. In case of stereo camera calibration "*camera matrix*", "*distortion matrix*", "*rectification matrix*", and "*projection matrix*" need to be determined. Once the calibration is done the calibration result is loaded to both the cameras. For the generation of the disparity image another ROS package "*stereo_image_proc*" is used. This package subscribes images from two camera drivers at the same time stamp and will generate the disparity image in "*stereo_msgs/DisparityImage.msg"* format as shown figure 3.2.

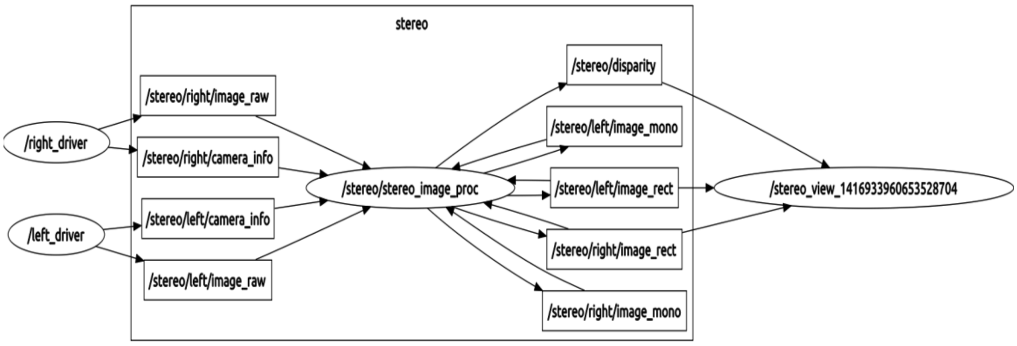
std_msgs/Header header	time-stamp in the header is the acquisition time of the first ray in the scan
sensor_msgs/Image image	generated disparity image
float32 f	focal length of the cameras
float32 T	base line of the stereo pair
sensor_msgs/RegionOfInterest valid_window	sub window of valid disparity values
float32 min_disparity	minimum disparity value possible
float32 max_disparity	maximum disparity value possible
float32 delta_d	smallest possible depth range

Fig 3.2. stereo_msgs/DisparityImage.msg" format

Generated disparity image will be available on "stereo/disparity" topic. *"_approximate_sync*" parameter was set to true to synchronize the residual time stamp mismatch due to the processing unit. Generated disparity image can be displayed along with the left and right images by the help of another ROS package "*image_view*" as shown in figure 3.3(a). Inter node connection during the disparity image generation is shown in figure 3.3(b).



(a)

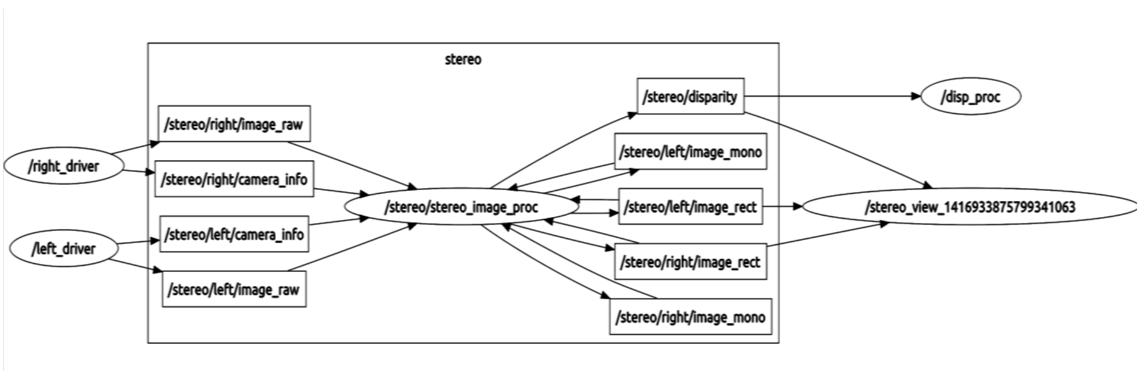


(b)

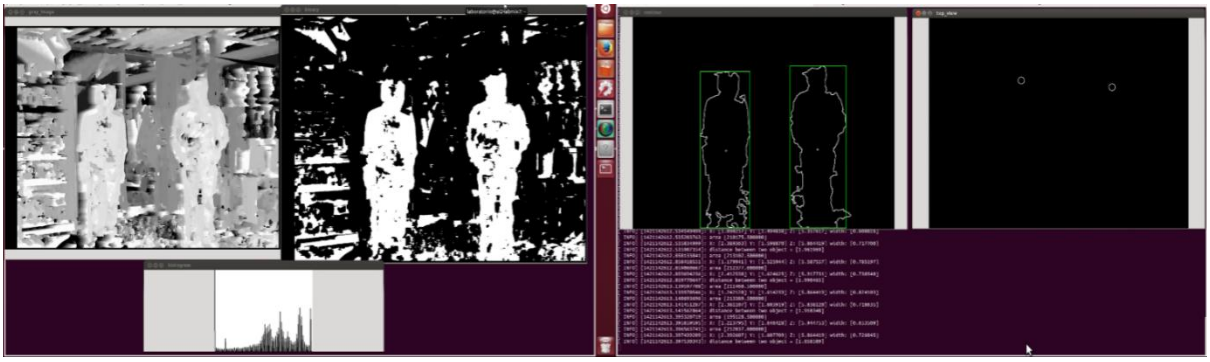
Fig 3.3. (a) generated disparity image (b)Inter node connection during the disparity image generation

Processing of the disparity image

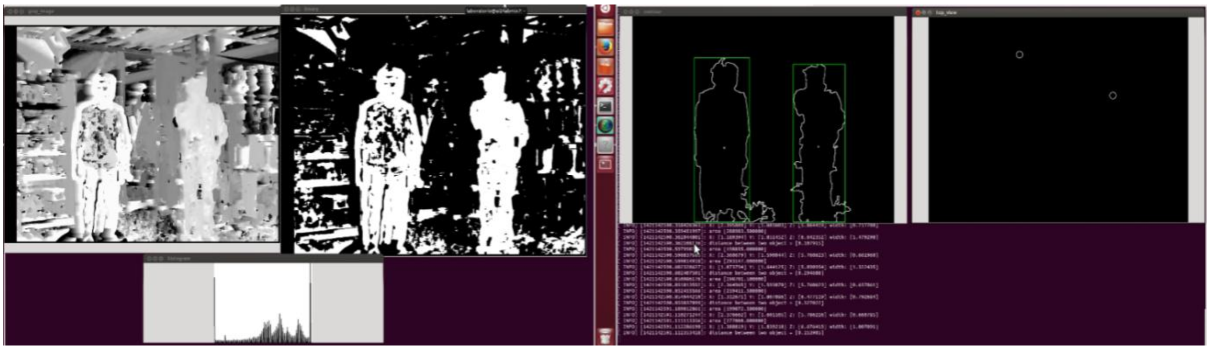
A new software tool "*disp_proc*" is developed for processing and extraction of information from the generated disparity image. "node" is responsible for the subscription of disparity image from the "*stereo_image_proc*", filtering the unwanted background and floor ground, detection of objects or pedestrians and finally calculation of object information like height, width, distance between two consecutive objects (if they belong to same depth level) and 3D coordinate with respect to the left_camera coordinate frame. For separation of background "watershed algorithm" (Based & Retrieval, 2001) is used and by inverse use of "3D-to-2D projection" 3D coordinate of the detected object or pedestrian is calculated. "*disp_proc*" is responsible for the generation of map view of different obstacles according to generated real world 3D coordinate. Inter node connection and generated map view with intermediate processing step is shown in figure 3.4. During map generation two different scenarios were considered. In the 1st case pedestrians are almost at same depth level and in the 2nd case pedestrians are at different depth level.



(a)



(b)



(c)

Fig 3.4. (a) Inter node connection during the "disp_proc" package execution (b) generated map view with pedestrians at same depth (c) generated map view with pedestrians at different depth

3.4. Task performed by laser

Laser scanner was used to implement the modified Breitenberg algorithm. Breitenberg algorithm is basically behaviour based Artificial Intelligence in a form of Sensorimotor interaction between the agent and its environment. Without any storage of data representation of the current environment can be done (Braitenberg, 1984). In the proposed method 2D laser scanner is used as a sensor module and wheel motors are used as effectors module. It means the presence of the obstacle along with their position will be sensed by the 2D laser scanner. According to the sensed data current angular velocity as well as linear velocity will be affected in order to avoid any collision. A new software tool is developed for the subscription of the laser data and performing the above mentioned task.

3.5. Summary

Camera calibration and stereo image (disparity image) generation was done using the existing software tools. A new software tool was developed to perform the image processing and feature extraction of the detected obstacles from the generated disparity image. Dynamic obstacle detection and laser scanner data was processed using the Breitenberg algorithm.

CHAPTER 4

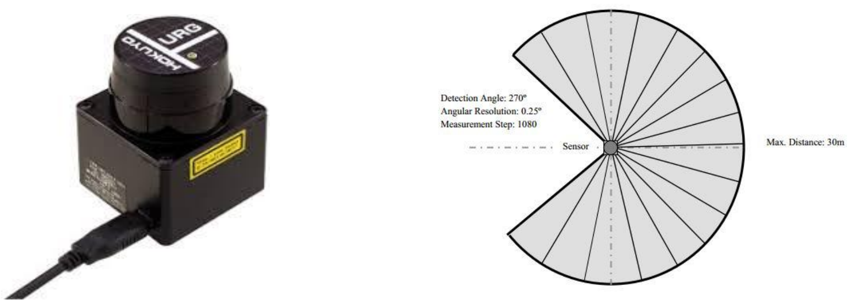
4. Materials and software

4.1. Overview

To perform the real-time tests for the developed algorithms we need corresponding hardware and software tools. For laser data Hokuyo Laser scanner was used and for stereo vision pair of Prosilica Camera was used. The detail description of the sensors and their integration with the robot platform is explained in this chapter. Summit_XL (product of Robotnik) was used as a robot platform. Robot Operating System (ROS) was used as a major software tool.

4.2. Laser scanner

In order to read the measures for automatic localization and robot navigation, the Hokuyo URG-04LX-UG01 as sown in figure 4.1 has been used. It is a small, affordable and accurate laser scanner very appropriate for robotics applications. The light source of the sensor is an infrared laser, and the scan area is a 240° semicircle with a maximum radius of 5600mm, with 1mm resolution. The angular resolution is 0.36° and sensor outputs the distance measured at every point (maximum: 684 steps). Laser beam diameter is less than 20mm at 2000mm with maximum divergence 40mm at 4000mm.



(a) *(b)*

Fig 4.1. (a) Hokuyo laser range finder (b) sensor scan area [26]

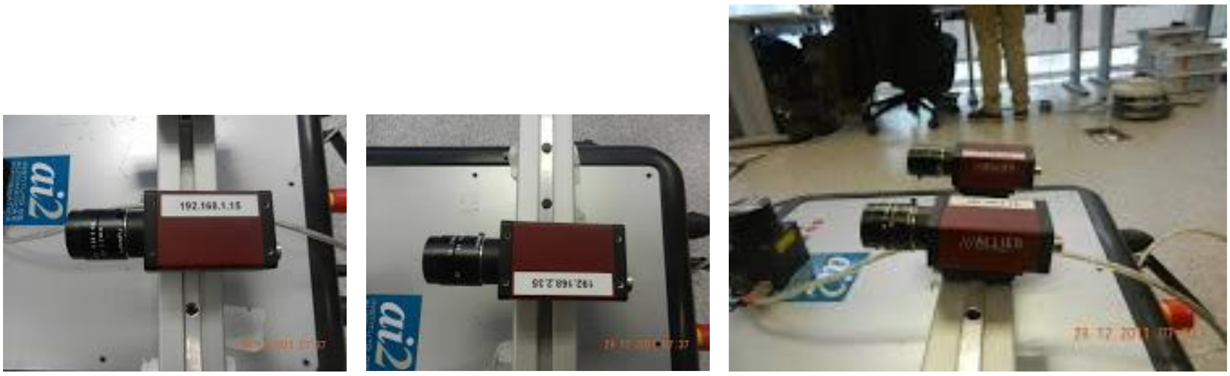
Principle of distance measurement is based on calculation of the phase difference, due to which it is possible to obtain stable measurements with minimum influence from object's color and reflectance. Normally the distance is calculated using the amount of time for the round trip between source and destination. In order to make the result more accurate and independent of the object color or reflectance, the phase difference method is incorporated according to equation 4.1.

$$D = \frac{c(N\pi + \Delta\Phi)}{4\pi f} \tag{4.1}$$

Where D is the calculated distance, c is the speed of light, f is frequency of the optical wave, N is the (integer) number of wave half cycle of the round trip and $\Delta\Phi$ is the rest of the phase shift.

4.3. Stereo camera

In order to generate the stereo vision two Prosillica cameras with Ethernet address 192.168.1.15 (left camera or master camera) and 192.168.2.35 (right camera or slave camera) are placed side by side with known baseline distance (gap between two cameras) shown in figure 4.2. It is a CCD based camera with resolution of 6576 x 4384. Power requirement and consumption are 7–25 VDC and 6.6W @ 12 VDC respectively. Body dimensions are 96 x 66 x 53.3 (including connectors, w/o tripod and lens) with 372g weight. The camera works within 20°C to +50°C temperature range.



(a) *(b)* *(c)*

Fig 4.2. (a) left camera (b) right camera (c) robot platform with stereo setup

Basic principle for the use of stereo vision is comparing individual pixel of the left and right images taken from both the cameras at the same timestamp (Priyanka et al., 2014). As the stereo vision works on a pixel basis algorithm it causes a huge data processing cost. At the same time single camera causes less processing cost. Stereo vision is used due to the following reasons:

- ✓ It facilitates 3D coordinate generation.
- ✓ Generates disparity information at each pixel.
- ✓ It is possible to segment the generated disparity image at different depth level.
- ✓ In stereo vision object detection can be done using depth information which is a common feature for any kind of object.

The system, therefore, is capable of detecting any kind of object irrespective of its shape, size, texture, colour etc.

4.4. Summit_XL mobile robot

Summit XL mobile robot as shown in figure 4.3(a) is used in this work.

(a) *(b)* *(c)*

Fig 4.3. (a) Summit_xl robot platform (b) internal processer (c) connections

The Summit XL robot has been developed by the company Robotnik. It is a mobile platform in differential configuration based on four independent brushless motors. It has a weight of 30 kg, a maximum load of 20kg extra, has a top speed of 3m/s and has autonomy of several hours. Its brain is composed of a mini ITX board with a CPU Intel Atom N2800 1.86GHz processor, 2GB DDR3 memory and a hard disk of 128GB solid state as shown in figure 7(b). It offers connections such as USB, Ethernet, VGA and HDMI ports and also includes a router configured as an access point to connect to the robot via WiFi as shown in figure 7(c). The operating system that comes standard is a Linux Ubuntu LTS. Moreover, the robot is factory set to be fully integrated into ROS.

4.5. Robot Operating System

ROS (Robot Operating System) stands out for being based on a completely modular, distributed architecture shown in figure 4.4, and have a very active community of developers. It shares some characteristics of other robotics software frameworks like Player, YARP, Orocos, CARMEN, Orca, MOOS o Microsoft Robotics Studio. Nevertheless ROS is fully open-source software that provides the ability to any developer to share their contributions to the community. Thus researchers are not constantly “reinventing the wheel”, but rather their research can be developed from previously tested and verified algorithms.

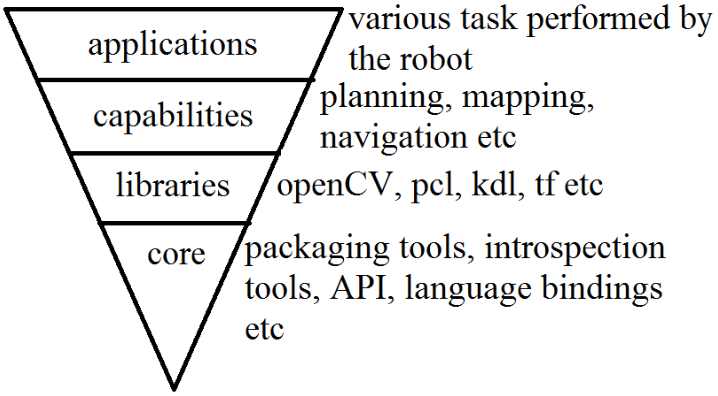


Fig 4.4: architecture of Robot Operating System (ROS)

The ROS project was initiated at Willow Garage in collaboration with the Stanford University and the University of California and defined as "an open-source meta-operating system". It presents the functionalities found in computer OSs but adapted towards a robotic system. Some of these are for example hardware abstraction, low-level device control, standard services and peer-to-peer communication. ROS supports programming in C ++, Python, Octave, LISP and even JAVA. It is developed under *Creative Commons Attribution* 3.0 license, which allows free use, distribution, and modification of all the code with the condition that they always specify the original developers. In addition, ROS integrates additional development tools like

***rviz*:** simulation of complete robots and environments with maps

rxgraph: visualization of node interconnection,

rosbag extreme useful data logging utility

rqt: framework for GUI development of ROS

rqt_reconfigurable: manipulate the parameters dynamically

There are many environment variables that can be set to affect the behavior of ROS. Of these the most important variables are

ROS_MASTER_URI: tells a node where master is.

ROS_IP and ROS_HOSTNAME: affect the network address of the node.

ROS_NAMESPACE: allows changing the namespace.

ROS_LOG_DIR: set the directory where the log files are written.

ROS_PACKAGE_PATH: list of all paths where ROS packages are resides.

It's really comfortable to set the environment variables in the .bashrc file with the help of source command. So there is no need to set all the variables every time a new terminal is being opened.

ROS takes the form of a conglomeration of various tools which are organized in packages. Packages are the smallest individual software organization unit of ROS code. This allows for easier installation and management of upgrades, which is the key to successful evolution of the project. Each package may contain libraries, includes, executables or src, YAML, CMakeLists and a manifest.

CMakeLists: The file CMakeLists.txt is the input to the CMake build system. This mainly contains name of the package, the depended packages and the list of the executables or nodes present in the package. The nodes mentioned in this CMakeList will only be taken care at the time of building of the package.

Manifest.xml: Manifest is a description of a package. It serves to define the dependences between packages and to capture the meta-information about the package like version, maintainer, license etc.

Executables or src: This folder contains all the executables or nodes present in the package as listed in the CMakeList.

Includes: This folder contains all the header files (if any)

YAML: All the parameters can be specified in the YAML file.

Node: A node is an executable that uses the ROS client library to communicate over the ROS network with other nodes. These nodes are mostly written in C++ or python language. A single package can support multiple nodes even written in different languages. The information exchange between nodes can be either by passing messages or using (requesting) services. They have a unique name to avoid possible misconnections. Nodes can publish messages to the topic or subscribe to it to receive messages from other nodes.

Master (roscore): The communication handshake is established by the master (*roscore*), which acts as a master that governs all the connections. Despite starting and ending all connections between nodes as shown in figure 4.5, *roscore* does not see the actual messages passing through as in a conventional router. Additionally, *roscore* is responsible for *rosout* (ROS stdout/stderr) and the parameter server (collection of network-shared node variables).

Publisher: Publisher is a class. In order to publish a message an object of that class must be created by means of node_handle, message_type and a specific topic_name.

Node_handle: is an object of class ros::NodeHandle which contains the advertise method. This method is actually responsible for publishing messages on the topic.

Message_type: is formally called the template parameter which is the data type for the message we want to publish.

Topic_name: the topic name is a string containing the name of the topic which we want to publish.

Subscriber: Publisher is a class. In order to subscribe a message an object of that class must be created by means of node_handle, queue_size and a specific topic_name.

Callback function: CallbackFunction is a dynamically executable sub-routine, which is called by the subscriber. It is implemented as a parameter of the subscriber. Every time a new message is available and received by the subscriber it will be executed.

Queue_size: is the integer size of the message queue for this subscriber. The purpose of this buffer is to avoid the loss of the messages by storing them. The loss of the messages may occur due to the slow execution order of the subscribing node.

The publisher and subscriber are working on a different layer. Due to its different execution layer the main execution order is not being hampered.

Messages: A message file consists of two fields name and type. From the message name the message type can be derived. The type of the message consists of the package name and the name of the message. For eg. geometry_msgs/msg/Twist.msg -> geometry_msgs/Twist.

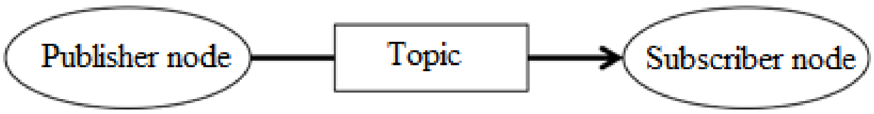


Fig4.5: generalized inter-node connection

Service and client:

Service calls are alternative method for communication in between different nodes as shown in figure 4.6. Even though the messages and services are transporting information between the nodes still there are fundamental differences.

1. The service calls are bi-directional. After sending the information service calls are always waits for a response. In contrast the publisher does not require any subscriber to publish a message.
2. The service calls are always one-to-one communication. The communication takes place

always between two particular nodes. In contrast a publisher can have any number of subscribers.

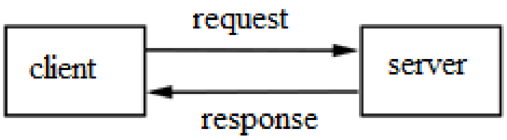


Fig 4.6: generalized client-server model

The service communication is a client server communication. The client node sends particular information as a request to the server node. The information can be configuration data parameters or actions. These ROS service calls are defined in a .srv file. The service files are similar to the message files except they consists of two parts a request and a response.

Launch file: ROS programs can be very extensive conglomerate of multiple packages, nodes, parameters, configuration files and data packages. To simplify the launch process ROS supports a quick launch mechanism by means of a single launch file. Launch file includes all the required elements as mentioned before.

4.6. Summary

Detail description of individual sensor, their driver packages available in the ROS, integration technique of those, and finally the important tools of the middleware is explained.

CHAPTER 5

5. Experimental setup

5.1. Overview

The procedures for the real-time test along with the setup description related to both the used sensors are explained in this chapter. Required message formats and modification related to packages pseudo code is also explained.

5.2. Setup for stereo test

The prosilica camera mainly needs two connections one for the power supply and other one are for data transmission. Power transmission is being done by standard 12V power cable and data transmission is being done by Ethernet cable. For software integration, a ROS package (*"prosillica_camera"*) has been used. It provides the basic methods of connection and receiving raw image data. . Generally in stereo vision two cameras will work in master-slave mode. One camera is considered as master (left camera) and other one (right camera) is slave. In order to have proper synchronization we have to run the master camera in fixed rate mode at about 10Hz frequency and the slave camera in External Sync node. It means the master camera will grab the image and simultaneously it will send a pulse to the slave camera. After receiving the pulse from master camera slave camera will grab the image. In order to launch both the cameras in the above mentioned modes a launch file is created. In the launch file Ethernet addresses of both the cameras and all the mode information against both the addresses are mentioned. All the provided information in the launch file are provided in the table 5.1. Without launching individual camera driver packages by launching the generated launch file the stereo vision mode can be achieved.

Table 5.1. Specifications of both the cameras mentioned in the launch file

Camera	ip_address	trigger_mode	frame_id
Left	192.168.1.15	Fixed rate	Map
Right	192.168.2.35	syncin1	Map

Due to module based structure individual sensors were tested. It means only stereo vision was configured and placed at top of the platform. Due to heavy processing cost of stereo matching algorithm stereo vision module was executed by the PC. The execution environment was transmitted to Summit_XL using wifi connection. The processed image result will be displayed on the screen of the PC. Since ROS supports distributed architecture, stereo vision module was even divided in two sub module to make the real time processing faster. Basic disparity matching was done by the Graphics Processing Unit (GPU), and rest of the post processing on the generated disparity image was done using Central Processing Unit (CPU). “l_image”, “r_image” and “disp_image” are created image matrix which are supported by the CPU unit. “d_left” “d_right” and “d_disp” are image matrix which are supported by the GPU unit. Before running the “block_matcher” images need to be uploaded to the GPU compatible image matrix and after the processing the generated disparity image need to be again download to the CPU compatible image matrix for further processing. The modifications made in the disparity matching code block are shown in figure 5.1.

```
const cv::Mat_<uint8_t> l_image = cv_bridge::toCvShare(l_image_msg, sensor_msgs::image_encodings::MONO8)-
>image;
const cv::Mat_<uint8_t> r_image = cv_bridge::toCvShare(r_image_msg,
sensor_msgs::image_encodings::MONO8)->image;
cv::Mat_<float> disp_image(disp_msg->image.height, disp_msg->image.width,
reinterpret_cast<float*>(&disp_msg->image.data[0]),
disp_msg->image.step);

cv::ocl::oclMat d_left, d_right;
cv::ocl::oclMat d_disp(disp_image.size(), CV_8U);
d_left.upload(l_image);
d_right.upload(r_image);
block_matcher_(d_left, d_right, d_disp);
d_disp.download(disp_image);
```

Fig 5.1. Modification made in the “stereo_image_proc” package

In order to check the reliability of the detection code pedestrian follower as well as avoider was tested. If the platform can able to generate the coordinate of the detected pedestrian and orient itself towards it then by simply inversing the logic it will definitely avoid. Experimental setup with the stereo vision is shown in the figure 5.2.

Fig 5.2. Stereo module with Summit_XL

5.3. Setup for laser test

The Hokuyo laser range finder is connected via USB to the controller unit of the robot. For software integration, a ROS package (*"hokuyo_node"*) has been used. It provides the basic methods of connection and receiving laser measurements. The connection is made using the COM port assigned to the laser, and measures are asynchronously received with a period of 400 ms per event. At each event 684 measures are stored and processed. Once a successful connection has been established the node will start publishing the data on a dedicated *"/scan"* topic using message "Sensor_msgs/*LaserScan.msg"*. Figure 5.3 shows the format of this message.

Header header	time-stamp in the header is the acquisition time of the first ray in the scan.
float32 angle_min	start angle of the scan [rad]
float32 angle_max	end angle of the scan [rad]
float32 angle_increment	angular distance between measurements [rad]
float32 time_increment	time between measurements [seconds]
float32 scan_time	time between scans [seconds]
float32 range_min	minimum range value [m]
float32 range_max	maximum range value [m]
float32[] ranges	range data [m]
float32[] intensities	intensity data [device-specific units]

Fig 5.3. Sensor_msgs/LaserScan.msg format

After successful orientation test laser module was tested on Summit_XL in indoor unstructured environment. After the stereo vision test was performed some new obstacles were added to make the environment unstructured. It means the new obstacles were not present during the stereo vision test and were unknown to Summit_XL during the Laser module test. Experimental setup with the Hokuyo laser is shown in the figure 12.

5.4. Summary

Detail of the laser data and disparity image message format along with their integration techniques are explained. Pseudo code responsible for the distribution of disparity processing load among the CPU and GPU is also provided.

CHAPTER 6

6. Procedures and Parameter Evaluation

6.1. Overview

Algorithms responsible for processing the data from laser and camera and finally use those data to control the movement of the mobile robot are explained in this chapter. Block matching technique for disparity image generation and extraction of obstacles from the image is explained in the first section of the chapter. Second section explained the modified Breitenberg algorithm and its advantages.

6.2. Disparity processing and orientation of robot

An algorithm is developed for the stereo module for detection, extraction of information about the detected objects and finally orienting the car towards the detected obstacles. The description of the algorithm is as follows.

1. Subscribe the generated disparity image
2. Convert the image to grey scale
3. Apply watershed algorithm to segment the background and obstacle layer
4. Extract the contour of the detected obstacle or pedestrian form obstacle layer
5. Place bounding box around the detected obstacle pedestrian
6. Find coordinate of the centroid of the contour in pixel unit
7. Create another bounding box (20 pixel X 20 pixel) considering contour centroid location as the centre
8. Calculate the average grey value within the box from the grey scale image generated in 2nd step.
9. Calculate depth or Z coordinate of the object from the gray value-depth map.
10. Calculate X and Y coordinate in the real world frame using Z, base line and focal length of the camera
11. Store the X, Y, and Z value in a vector
12. If contour portion present at the bottom left or bottom right region of the box
    a. Consider it as floor ground and ignore
13. Calculate the width of the bounding box in pixel unit
14. Convert the pixel result to real world unit using focal length and baseline value.
15. Repeat the 12th step for each pixel in the bounding box in a column wise manner

16. Repeat 6th step for each detected contour
17. Plot the detected obstacles in a new frame using the stored X and Z value.
18. Calculate the angular rotation required in order to orient the robot towards the obstacle.
19. Supply the angular velocity (sign same as X coordinate) to the robot.
20. Subscribe the current robot odometry information
21. If amount of rotation performed < amount of rotation required
    a. Go to 19th step
22. Goto step 1

Generated disparity image basically holds the amount of shift of a point between the left image and the right image. The shift value is known as disparity value. With disparity value depth or distance of the point from the camera can be determined using equation (6.1) shown below.

$$Z = \frac{f}{D} B \tag{6.1}$$

Where, Z is distance from the camera in m, f is focal length of the camera, D is the disparity value of the co-responding pixel, B is the base line value of the stereo setup. Total disparity range is 0 to 64 which is scaled with the total grey value range (0 to 255) at the time of grey scale conversion. So that after all the post processing only with the grey value at any point the disparity value can be estimated.

Estimation of the centroid is done using the *"centroid()"* function in OpenCV library. The function returns the centroid of the detected contour (u, v) in pixel unit considering extreme bottom left corner as origin. Generated 2D coordinate is transformed to camera frame considering centre of the frame as a new origin using equation 6.2 and 6.3 for x and y respectively.

$$u = \frac{x}{dx} + C_X \tag{6.2}$$

$$v = \frac{y}{dy} + C_Y \tag{6.3}$$

Where, u and v are the centroid location height and width (pixel unit), x and y are the coordinate in the camera frame, C_X and C_Y are new origin location in the camera frame. dx and dy are the dimension of each pixel in camera frame (can be found in the camera specification). Equation 3 and 4 can be rewritten as equation 6.4 and 6.5 respectively shown below.

$$x = (u - C_X)dx \tag{6.4}$$

$$y = (v - C_Y)dy \tag{6.5}$$

After the transformation of x and y coordinate in the camera frame, distance or Z_R value needs to be calculated. In order to determine the depth we need to know the corresponding grey value at that point. To make the depth estimation more accurate instead of relying only on the coordinate point a box is created and average pixel value is calculated from the grey scale image. Once the average grey value is found, using the disparity-grey scale corresponding average disparity was calculated. The calculated average disparity is considered as the disparity of the whole object. Disparity value is used in equation 2 to calculate the depth or Z_R. With known Z_R, x and y values real world coordinate generation technique is shown in figure 6.1.

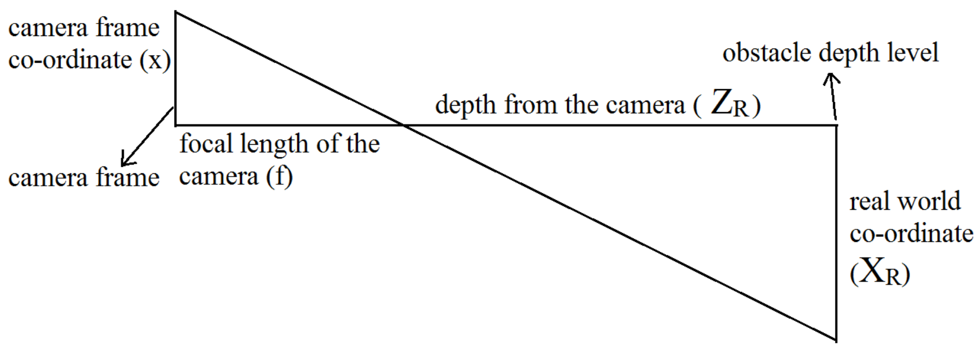


Fig 6.1. Camera frame to real world coordinate generation for x coordinate

From figure 13 the values of real world x-coordinate (X_R) and y-coordinate (Y_R) can be obtained as given in equation 6.6 and equation 6.7 respectively.

$$X_R = \frac{x}{f} * Z_R \tag{6.6}$$

$$Y_R = \frac{y}{f} * Z_R \tag{6.7}$$

During the calculation of the required angular rotation only x coordinate location and the depth or distance of the object form the robot is used. The calculation method is explained in figure 6.2.

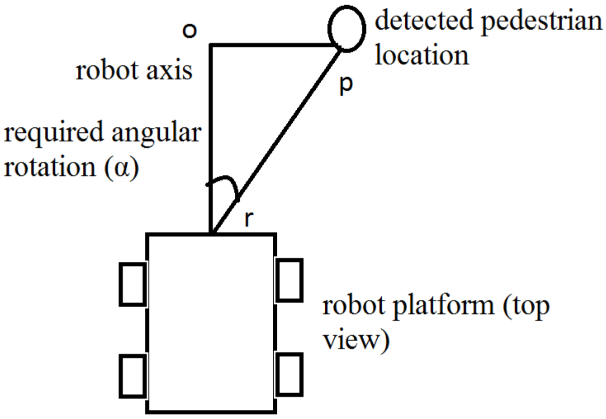


Fig 6.2. Robot orientation calculation towards detected obstacle

In order to estimate the amount of robot rotation (α) required to orient it towards the pedestrian, figure 14 can be used to deduce the following equations (equation 6.8 and equation 6.9).

$$\tan(\alpha) = \frac{OP}{OR} \tag{6.8}$$

$$\alpha = \tan^{-1}\frac{OP}{OR} \tag{6.9}$$

6.3. Modification of Breitenberg algorithm

The algorithm used by the robot for avoidance of sudden obstacles or pedestrian is as follows,

1. Subscribe laser data set from the hokuyo driver package
2. Log the selected angular laser scan data.
3. If scan value = inf or scan value > 5 or <0.001

a. Scan value = 0

4. Repeat the 3rd step for all the scan values
5. Multiply the scan values with the corresponding weightage values
6. Add the multiplied values and divide it with a constant
7. Supply the calculated values to the robot as angular velocity.
8. Go to step 1

Available scan range data of the laser scanner is from -120 deg to +120 deg with an angle resolution of 0.36 deg. Among these huge data set only -35 deg to +35 deg has been selected with an angle resolution of approximately 5 deg shown in figure 15. Selection and resolution value was decided considering maximum width of the robot platform and minimum width of obstacle possible respectively. More weightages are supplied to the scan range data at angle nearer to central axis and lesser weightages are supplied to the scan range data at angle nearer to outer axes. Angular sample locations and corresponding weightages values are shown in table 6.1. The obstacle almost within the path will cause a higher angular velocity and the obstacle at the border of the path will cause a lesser angular velocity. Linear and angular velocity estimation is done using equations 6.10 and 6.11 respectively.

$$vel_{ang} = \frac{1}{k_a} \sum W_i m_i \tag{6.10}$$

$$vel_{lin} = \frac{1}{k_l} \left|\sum W_i m_i\right| \tag{6.11}$$

Where k_a, k_l are positive constants, W is the weightage vector and m is the laser measurements vector.

Table 6.1. laser scan samples and corresponding weightages

Sl. No.	Sampled scan angel (deg)	Data step number	Weightage value
1	77	220	1.75
2	81	232	3.05
3	89	255	3.45
4	96	275	4.65
5	101	288	5.05
6	105	301	6.5
7	109	312	7.7
8	113	323	9.2
9	117	334	10.2
10	121	345	-10.0
11	124	356	-9.0
12	128	367	-7.5
13	132	378	-6.4
14	137	391	-5.0
15	141	404	-4.6
16	145	416	-3.4
17	151	431	-3.0
18	156	445	-1.7

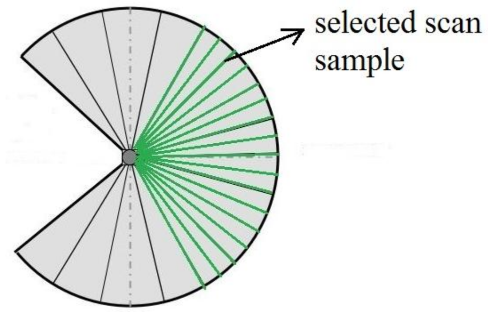


Fig 6.3. Selected scan samples in the total scan region

6.4. Basic advantages for the use of Breitenberg algorithm

1. If there is an obstacle within the path of the vehicle then only it should make the turn otherwise it will follow the current course generated by the stereo vision module. So no unwanted movement
2. If the obstacle is in the right portion of the path the robot will make a left turn and vice versa in order to avoid unnecessary extra turn.
3. Reduction in extra movement will reduce the total travel path and make the system more goal orienting its means amount of deviation from the current course is minimum possible.
4. Due to the reduction in the data set size processing time as well as cost will be reduced. Reduction in processing time is very much important for dynamic unstructured obstacles.

6.5. Summary

Techniques for detection of obstacles as well as pedestrians, calculation of the centroid, and finally orientating the mobile robot towards it are explained. Modifications made on the Braitenberg algorithm and their corresponding advantages are also explained.

CHAPTER 7

7. Result

7.1. Overview

After successful development of corresponding algorithms it needs to be checked in the real time. For real time test previously mentioned hardware were used. Due to its modular architecture, tasks of individual sensors were tested installing them on the mobile platform.

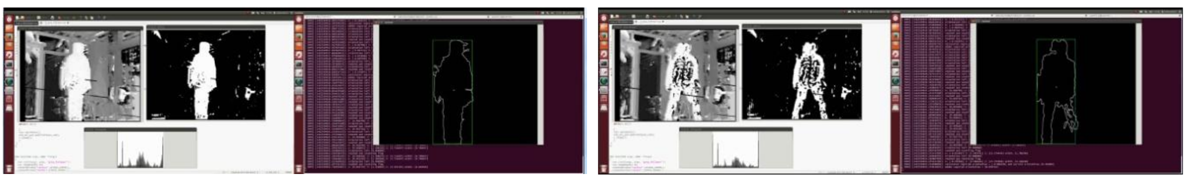
7.2. Result of stereo image processing

Indoor test on Summit_XL was performed only with the stereo module attached to it. Generated disparity image and intermediate processing steps were shown in figure 3.4. After successful detection the robot orients itself towards the human. The real world test results and corresponding robot vision are shown in figure 7.1 and 7.2 respectively. Both the test videos are available at the following links respectively:

https://www.youtube.com/watch?v=sicpWKTOe78

https://www.youtube.com/watch?v=iY5XZq3sVMs&feature=em-upload_owner

Fig 7.1. Test result of human follower with Summit_XL



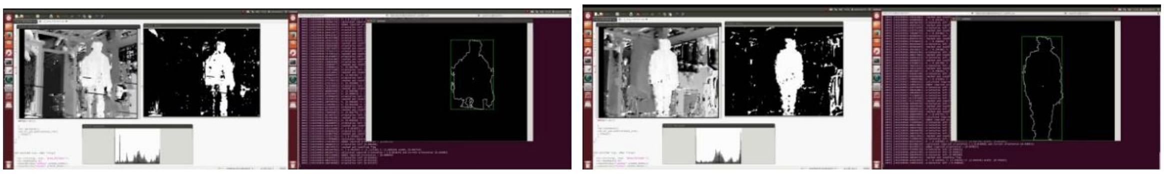


Fig7.2. Robot vision during human follower

7.3. Result of Modified Breitenberg algorithm

Wooden boxes were placed in front of the car after the performance of stereo block to make them unstructured. Successful simulation test in Gazebo simulator (ROS) and indoor test run are shown in figure 7.3 and 7.4 respectively. Both the test videos are available at the following links respectively:

https://www.youtube.com/watch?v=j4N4djLH-3E&feature=em-upload_owner

https://www.youtube.com/watch?v=NiZ_KoggfIs&feature=em-upload_owner

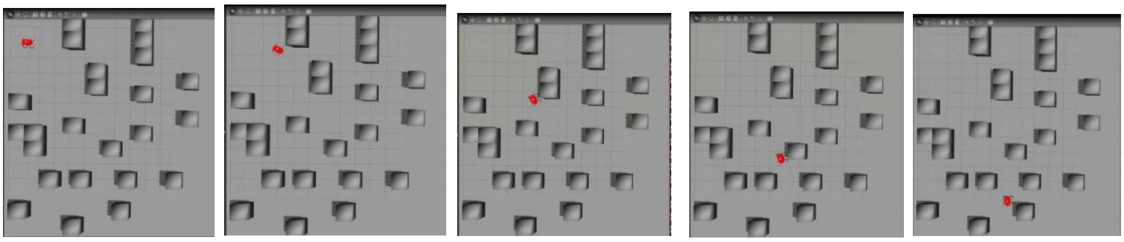


Fig 7.3. Breitenberg simulation result using Gazebo simulator

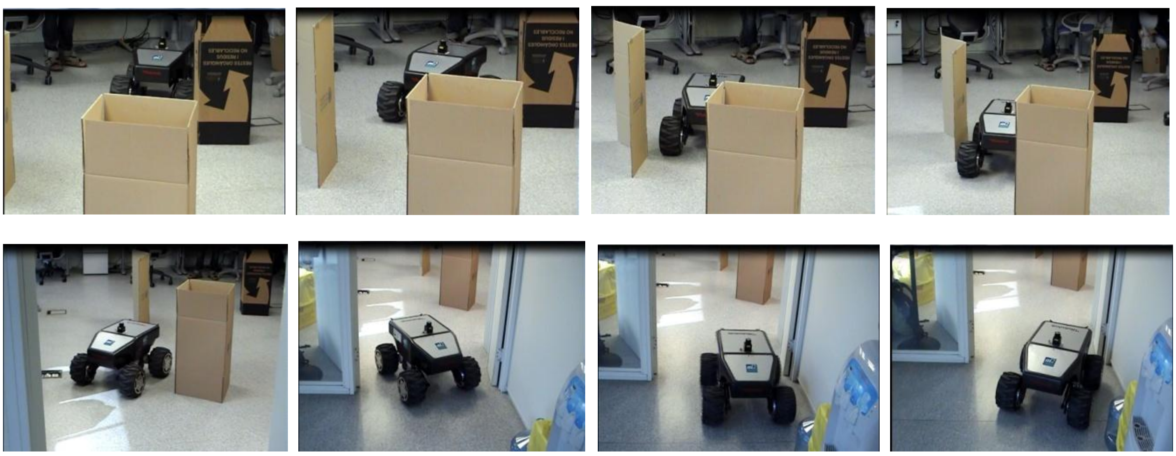


Fig 7.4. Breitenberg indoor test with Summit_XL

The angular velocity response of the vehicle during the presence of an obstacle is shown in figure 7.5. Along with the central axis sample other 18 samples on both the sides equally divided are used to implement the Braitenberg algorithm. One sample from left side, one sample from right side and the central axis sample are chosen to explain the plot. First plot is the variation of the angular velocity. Until the scan range data are greater than 5 meters, angular velocity remains 0 rad/sec. More it moves nearer to the obstacle the angular velocity goes on increasing. The time it starts avoiding the obstacle the angular velocity again starts reducing and finally comes to 0 rad/sec. Rest of the plot explains the variation of laser scan data for left side sample, right side sample, and central axis sample respectively. In this case the vehicle takes a right turn. So the left side sample reaches to its maximum value later than the central axis sample and right side sample. The central axis sample reaches to its maximum value later than the right side sample but before left side sample

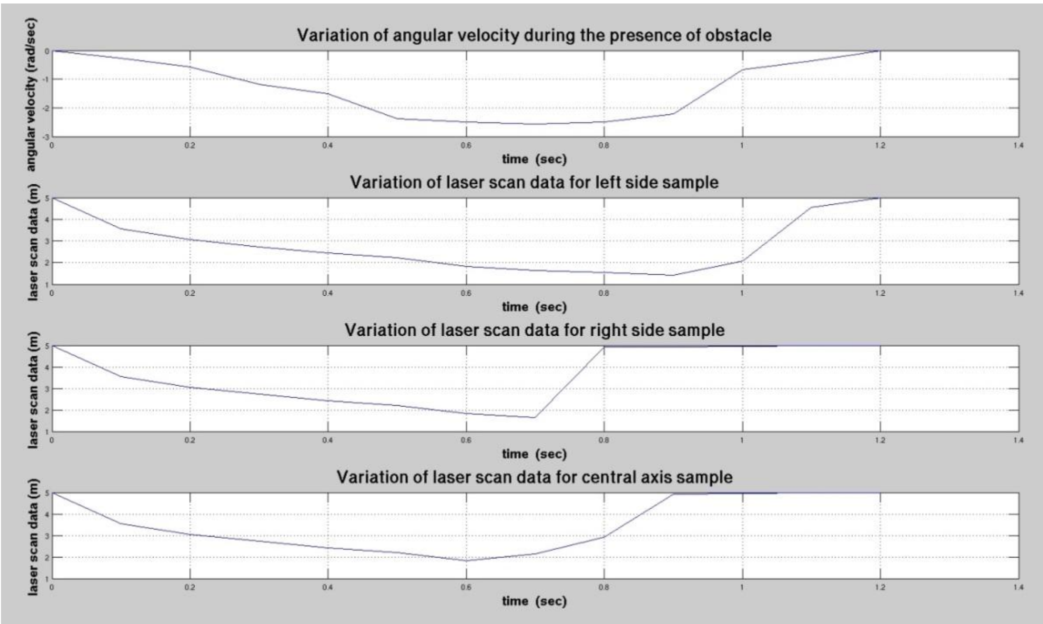


Fig 7.5. Breitenberg collision avoidance response

7.4. Summary

Successful hardware tests were performed. Due to limitation in the hardware only indoor tests were performed. The test result videos were also uploaded at the internet. Addresses were provided in the previous section.

CHAPTER 8

8. Conclusions and future work

8.1. Conclusions

This paper has presented a solution to the autonomous navigation for light vehicles problem using advanced sensors. Major key issues considered in the present work were generation of real-time disparity image generation due to its huge processing time, orienting the mobile robot towards the most available gap among all the obstacles including the pedestrians, and finally avoidance of the dynamic obstacles during the movement. Processing load of the disparity image generation was shifted to the GPU unit and rest of the post processing for calculation of gap among different obstacle was done by the CPU unit to make the system real-time. Avoidance of dynamic obstacles was done by the use of well tuned Breitenberg algorithm. The various tasks that make up the navigation algorithm have been identified in format software modules endowed with a certain ability to communicate between them, giving the system great flexibility, scalability and reusability. The work has shown the hardware/software platform used. The hardware is based on the Summit XL mobile robot equipped with a stereo vision setup and a laser rangefinder. The software control architecture developed is based on the robot programming framework of ROS. The paper has described the proposed solutions to various problems related with the autonomous navigation. These solutions have solved the indoor and outdoor navigation with obstacle avoidance. The results obtained with this platform are very satisfactory. Since the obstacle detection is done by the use of depth property any kind of obstacle could be detected. The amount of angular rotation the mobile robot would perform at the time of occurrence of dynamic obstacle is least possible which makes the system more goal-oriented.

8.2. Future scope

In future some modifications could make the proposed system smarter. The modifications envisaged may be the following

1. Use of properly tuned PID controller for the orientation of the robot during the follower to make the system smoother
2. Reducing the processing time of disparity image generation
3. Generation of better quality disparity image
4. More outdoor tests need to be performed

References

Aggarwal, A., Kukreja, A., & Chopra, P. (2010). Vision based collision avoidance by plotting a virtual obstacle on depth map. *2010 IEEE International Conference on Information and Automation, ICIA 2010*, 532–536. http://doi.org/10.1109/ICINFA.2010.5512394

Alonso, D., Pastor, J. Á., Sánchez, P., Álvarez, B., & Vicente-chicote, C. (2012). Generación Automática de Software para Sistemas de Tiempo Real : Un Enfoque basado en Componentes , Modelos y Frameworks, *9*, 170–181. http://doi.org/10.1016/j.riai.2012.02.010

Arrichiello, F., Chiaverini, S., & Mehta, V. K. (2012). Experiments of obstacles and collision avoidance with a distributed multi-robot system. *2012 IEEE International Conference on Information and Automation, ICIA 2012*, (June), 727–732. http://doi.org/10.1109/ICInfA.2012.6246771

Based, C., & Retrieval, I. (2001). 5 . Foreground Objects Detection & Background Separation, 76–107.

Bischoff, R., Guhl, T., Prassler, E., Nowak, W., Kraetzschmar, G., Soetens, P., … Leuven, K. U. (2010). BRICS – Best practice in robotics Summary / Abstract, 968–975.

Bruyninckx, H. (2001). Open Robot Control Software : the OROCOS project, 2523–2528.

Cheam, S., Bakar, A., & Saman, S. (2014). Developing Algorithm for Object Tracking Using Passive Sensors.

Chen, C. C., Lu, M. C., Chang, P. C., Tsai, C. P., & Tang, T. Y. (2009). Image-based detection and obstacle avoidance for mobile robots. *ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents*, 193–197. http://doi.org/10.1109/ICARA.2000.4803921

Chen, C. L., Chou, C. C., & Lian, F. L. (2011). Active pedestrian following using laser range finder. *2011 IEEE International Conference on Information and Automation, ICIA 2011*, (June), 690–695. http://doi.org/10.1109/ICINFA.2011.5949082

Cherian, a., Morellas, V., & Papanikolopoulos, N. (2009). Accurate 3D ground plane estimation from a single image. *2009 IEEE International Conference on Robotics and Automation*, 2243–2249. http://doi.org/10.1109/ROBOT.2009.5152260

Chi, K.-H., & Lee, M.-F. R. (2011). Obstacle avoidance in mobile robot using Neural Network. *IEEE International Conference on Consumer Electronics, Communications and Networks*, 5082–5085. http://doi.org/10.1109/CECNET.2011.5768815

Csaba, G., Somlyai, L., & Vamossy, Z. (2011). Mobile robot navigation in unknown environment using structured light. *3rd IEEE International Symposium on Logistics and Industrial Informatics*, 249–254. http://doi.org/10.1109/LINDI.2011.6031158

Danilo, C. (2014). Laser Based Obstacle Avoidance Strategy for Autonomous Robot Navigation Using DBSCAN for Versatile Distance.

Dur, E. (2009). Optical flow-based obstacle detection and avoidance behaviors for mobile robots used in unmaned planetary exploration. *RAST 2009 - Proceedings of 4th International Conference on Recent Advances Space Technologies*, 638–647. http://doi.org/10.1109/RAST.2009.5158270

Einhorn, E., Schröter, C., & Gross, H.-M. (2009). Monocular Scene Reconstruction for Reliable Obstacle Detection and Robot Navigation. *European Conference on Mobile Robots*, 156–161. http://doi.org/10.1016/j.robot.2011.02.008

Ess, a, Leibe, B., Schindler, K., & van Gool, L. (2009). Moving obstacle detection in highly dynamic scenes. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 56–63. http://doi.org/10.1109/ROBOT.2009.5152884

Essen, M. Von, Hirvonen, J., & Kuikka, S. (2014). Robotic software frameworks and software component models in the development of automated handling of individual natural fibers, 29–45. http://doi.org/10.1007/s12213-014-0078-8

Gerkey, B. P., Vaughan, R. T., & Howard, A. (2003). The Player / Stage Project : Tools for Multi-Robot and Distributed Sensor Systems. *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, (Icar), 317–323. http://doi.org/10.1.1.10.491

Gill, C. D., Smart, W. D., & Louis, S. (2002). Middleware for Robots ?

Guo, J., Su, K., Wang, C., & Wu, C. (2009). Laser Range Finder Applying in Motion Control System of Mobile Robots. *2009 Fourth International Conference on Innovative Computing, Information and Control*, 536–539.

Hoang, T. T., Hiep, D. T., Duong, P. M., Van, N. T. T., Duong, B. G., & Vinh, T. Q. (2013). Proposal of algorithms for navigation and obstacles avoidance of autonomous mobile robot. *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, 1308–1313. http://doi.org/10.1109/ICIEA.2013.6566569

Hotel, T. G. (2010). No Title, 2507–2510.

Huang, W. H., Fajen, B. R., Fink, J. R., & Warren, W. H. (2006). Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, *54*(4), 288–299. http://doi.org/10.1016/j.robot.2005.11.004

Jeong, H. K. J. H. K., Kim, S. H. K. S. H., & Kwak, Y. K. K. Y. K. (2009). Development of autonomous robot system for indoor messy environments. *2009 Iccas-Sice*, 3409–3413.

Kang, X., Zhu, W., Li, K., Tian, L., Zhang, M., & Jiang, J. (2012). The new adaptive clustering method of laser scanner data for automated vehicle obstacle recognition in unstructured environment. *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, 1154–1161. http://doi.org/10.1109/ICMA.2012.6283414

Karambakhsh, a., Yousefi Azar Khanian, M., Meybodi, M. R., & Fakharian, a. (2011). Robot navigation algorithm to wall following using fuzzy Kalman filter. *IEEE International Conference on Control and Automation, ICCA*, 440–443. http://doi.org/10.1109/ICCA.2011.6138043

Kim, S., & Kim, H. (2011). Systematic optimal design of overlapped ultrasonic sensor ring for high performance obstacle detection. *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics, ICAR 2011*, 600–605. http://doi.org/10.1109/ICAR.2011.6088544

Körtner, T., Schmid, A., Batko-klein, D., Gisinger, C., Huber, A., Lammer, L., & Vincze, M. (n.d.). How Social Robots Make Older Users Really Feel Well – A Method to Assess Users ' Concepts of a Social Robotic Assistant, 138–147.

Lin, Y. C., Lin, C. T., Liu, W. C., & Chen, L. T. (2013). A vision-based obstacle detection system for parking assistance. *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, 1627–1630. http://doi.org/10.1109/ICIEA.2013.6566629

Makarenko, A., Brooks, A., & Kaupp, T. (n.d.). Orca : Components for Robotics.

Moghadam, P., & Wijesoma, W. S. (2008). Improving path planning and mapping based on stereo vision and lidar. *2008 10th International Conference on Control Automation Robotics and Vision*, (December), 384–389. http://doi.org/10.1109/ICARCV.2008.4795550

Moon, H., Kim, J., & Kim, J. (2007). Obstacle Detecting System for Unmanned Ground Vehicle using Laser Scanner and Vision, 1758–1761.

Mukai, T., Hirano, S., Nakashima, H., Kato, Y., Sakaida, Y., Guo, S., & Hosoe, S. (2010). Development of a nursing-care assistant robot RIBA that can lift a human in its arms. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 5996–6001. http://doi.org/10.1109/IROS.2010.5651735

Nagatani, K., Kiribayashi, S., Okada, Y., Tadokoro, S., Nishimura, T., & Yoshida, T. (2011). Redesign of rescue mobile robot Quince – Toward emergency response to the nuclear accident at Fukushima Daiichi Nuclear Power Station on March 2011 –.

Obstacle, E., Module, A., Using, M. R., Finder, L. R., Development, I., & Urg-, H. (2013). Hideyuki Saito, Ryosuke Amano, Naruhito Moriyama, Kazuyuki Kobayashi, and Kajiro Watanabe, 348–353.

Perrollaz, M., Yoder, J. D., & Laugier, C. (2010). Using obstacles and road pixels in the disparity-space computation of stereo-vision based occupancy grids. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1147–1152. http://doi.org/10.1109/ITSC.2010.5625162

Priyanka, K. B., Deepu, R., Honnaraju, B., Tech, M., Prof, A., & Prof, A. (2014). GENERATION OF IMAGES FROM MONOCULAR IMAGE FOR STEREOVISION, *1*(11), 1295–1299.

Quigley, M., Conley, K., Gerkey, B., FAust, J., Foote, T., Leibs, J., … Mg, A. (2009). ROS: an open-source Robot Operating System. *Icra*, *3*(Figure 1), 5. http://doi.org/http://www.willowgarage.com/papers/ros-open-source-robot-operating-system

Ringer, R. J., Pharm, D., Wagner, T. H., Ph, D., Krebs, H. I., Ph, D., … Ph, D. (2010). Robot-Assisted Therapy for Long-Term Upper-Limb Impairment after Stroke.

Shaw, M., & Clements, P. (2006). The golden age of software architecture. *IEEE Software*, *23*(2). http://doi.org/10.1109/MS.2006.58

Shibata, T. (2012). Therapeutic seal robot as biofeedback medical device: Qualitative and quantitative evaluations of robot therapy in dementia care. *Proceedings of the IEEE*, *100*(8), 2527–2538. http://doi.org/10.1109/JPROC.2012.2200559

Son, H., Kim, C., & Choi, K. (2010). Rapid 3D object detection and modeling using range data from 3D range imaging camera for heavy equipment operation. *Automation in Construction*, *19*(7), 898–906. http://doi.org/10.1016/j.autcon.2010.06.003

Tang, J. F., Mu, L. F., Kwong, C. K., & Luo, X. G. (2011). An optimization model for software component selection under multiple applications development. *European Journal of Operational Research*, *212*(2), 301–311. http://doi.org/10.1016/j.ejor.2011.01.045

Toit, N. E. Du, & Burdick, J. W. (2012). Robot Motion Planning in Dynamic , Uncertain Environments, *28*(1), 101–115.

Voda, A. (2013). Four Driving / Steering Wheels Autonomous Vehicle, 187–192.

Wei, C., Zeng, Y., & Wu, T. (2013). Obstacle detection based on a 2D large range sonar model. *Proceedings of the 2013 6th International Congress on Image and Signal Processing, CISP 2013*, *2*(Cisp), 1127–1131. http://doi.org/10.1109/CISP.2013.6745225

Wijayasiriwardhane, T., Lai, R., & Kang, K. C. (2011). Effort estimation of component-based software development – a survey. *IET Software*, *5*(2), 216. http://doi.org/10.1049/iet-sen.2009.0051

Wu, Q., Gao, J., Huang, C., Zhao, Z., Wang, C., Su, X., … Xu, Z. (2012). Obstacle Avoidance Research of Snake-like Robot based on Multi-sensor Information Fusion, 1040–1044.

Yu, Z., Gao, M., Deng, X., Du, L., He, C., Di, J., … Hu, L. (2010). APF obstacle avoidance in polar coordinates for mobile robot based on laser radar. *2010 2nd International Conference on Advanced Computer Control*, 549–552. http://doi.org/10.1109/ICACC.2010.5487169

Zhao, X., Wu, N., Lou, Y., & Ogai, H. (2011). Development of automatic driving system by robocar. *SICE Annual Conference 2011*, (1), 2170–2173.

Zug, S., Penzlin, F., Dietrich, A., Nguyen, T. T., & Albert, S. (2012). Are laser scanners replaceable by Kinect sensors in robotic applications? *2012 IEEE International Symposium on Robotic and Sensors Environments, ROSE 2012 - Proceedings*, 144–149. http://doi.org/10.1109/ROSE.2012.6402619